

HTTP Browser Hints
draft-nottingham-http-browser-hints-03

Abstract

Over time, Web browsers have adapted how they use HTTP based upon common server configurations and behaviours. While this is necessary in the common case, it can be detrimental for performance and interoperability.

This document establishes a mechanism whereby origin servers can make available hints for browsers about their preferences and capabilities, without imposing overhead on their interactions or requiring support for them.

This is intended to allow browsers to safely optimise connections to servers.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 22, 2012.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
2.	Requirements	3
3.	A file format for Browser Hints	3
3.1.	The 'prefixlist' Value Type	4
4.	Discovering Browser Hints for a Web site	4
5.	Pre-defined Browser Hints	5
5.1.	max-conns	5
5.2.	pconn-ip	5
5.3.	ip-balance	6
5.4.	connect-timeout	6
5.5.	read-timeout	6
5.6.	max-pipeline-depth	7
5.7.	small-hdrs	7
5.8.	relative-referer	7
5.9.	chunk-req-bodies	7
5.10.	omit-cookies	8
6.	Security Considerations	8
7.	IANA Considerations	8
7.1.	The 'browser-hints' Well-Known URI	8
7.2.	The HTTP Browser Hints Registry	8
8.	References	9
8.1.	Normative References	9
8.2.	Informative References	9
Appendix A.	Acknowledgements	9
	Author's Address	10

Nottingham

Expires December 22, 2012

[Page 2]

1. Introduction

HTTP [[RFC2616](#)] clients -- especially browsers -- typically use hardcoded values or heuristics to determine how many TCP connections to use to a server, based on common-case server behaviours and limitations.

Likewise, they often send voluminous request headers (e.g., in User-Agent and Allow) because they fear that changing those headers' values will break some sites that depend upon specific values.

These are just two examples of common, conservative behaviour by browsers that is good for interoperability, but potentially bad for performance in certain circumstances.

This memo proposes a mechanism whereby a HTTP server can advertise hints for browsers (and other clients), so that communication with them can be optimised.

It does so by defining a file format for such Browser Hints [Section 3](#), and defining how clients can discover it for a given Web site [Section 4](#). Finally, an extensible vocabulary of hints is defined [Section 5](#).

Feedback for this draft should take place on the apps-discuss@ietf.org mailing list
<<https://www.ietf.org/mailman/listinfo/apps-discuss>>.

2. Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC2119](#)].

3. A file format for Browser Hints

Browser hints are indicating using a JSON [[RFC4627](#)] formatted file, containing a single object whose member's names are browser hints, as defined by the registry [Section 7.2](#).

For example;

```
{
  "max-conns": 5,
  "small-hdrs": true
}
```

Nottingham

Expires December 22, 2012

[Page 3]

By their nature, all browser hints are optional; i.e., browsers are free to ignore them.

3.1. The 'prefixlist' Value Type

Each browser hint is defined to have a JSON-derived value type; e.g., 'string' or 'array'. This section defines a special value type, 'prefixlist' that is an array containing one or more arrays, each containing a path prefix followed by either 'true' or 'false' to indicate whether the hint applies to that path.

Prefixlists are evaluated in order, with the first case-sensitive, character-by-character prefix match for a given URI's path determining whether the hint applies.

For example, given the following hint document:

```
{
  "omit-cookies": [
    ["/images/users/", false],
    ["/images/", true]
  ]
}
```

We can tell that "omit-cookies" applies to resources under the "/images/" path (such as "/images/123.jpg"), except for those under "/images/users/" (such as "/images/users/bob.jpg").

If a value specified to be a prefixlist is either 'true' or 'false' that indicates that the hint applies to the whole site, or does not apply to the whole site, respectively.

For example,

```
{
  "omit-cookies": true
}
```

Indicates that the "omit-cookies" hint applies to the entire site.

Prefixlists can only be used when the browser hint's registration nominates their use.

4. Discovering Browser Hints for a Web site

The hints relevant to a given site can be determined by fetching the URI path `"/.well-known/browser-hints"` for that site.

Nottingham

Expires December 22, 2012

[Page 4]

Typically, clients (especially browsers) will not block other requests to a site while fetching the browser hints (because they're optional); instead, it will usually be done concurrently with other requests, or on idle connections for future use.

In this specification, "site" is scoped by the URI scheme and authority. As such, all of the following are considered to be different sites, and therefore have different browser hints:

- o <http://foo.com/>
- o <https://foo.com/>
- o <http://foo.com:8000/>
- o <http://www.foo.com/>

Clients SHOULD follow HTTP 3xx redirects when retrieving hints.

A successful response is valid for its associated site for as long as it can be cached in HTTP.

If the response has a 200 status code but no explicit freshness (e.g., a Cache-Control: max-age or Expires: header), clients SHOULD cache the response heuristically for a generous fixed period (e.g., 14 days).

If the response has a 404 status code but no explicit freshness, clients SHOULD cache the response heuristically for a generous fixed period (e.g., 14 days).

5. Pre-defined Browser Hints

5.1. max-conns

- o Browser Hint Name: max-conns
- o Description: When present, this hint indicates the maximum number of concurrent persistent connections that the site would like clients to use.
- o Value Type: number
- o Contact: mnot@mnot.net

5.2. pconn-ip

- o Browser Hint Name: pconn-ip
- o Description: When true, this hint indicates that the site allows clients to reuse persistent connections keyed by IP address, rather than by hostname. Note that all sites that are sharing the connection MUST declare this hint for it to be used, and if a transport-layer certificate is in use (e.g., for TLS [[RFC5246](#)]),

Nottingham

Expires December 22, 2012

[Page 5]

- it MUST be valid for all sites.
- o Value Type: true | false
- o Contact: mnot@mnot.net
- o Specification: [this document]

In other words, if both `www.example.com` and `foo.example.org` resolve to the address `192.0.2.5`, and indicate this hint, then clients can send a request to `www.example.com` and then a request to `foo.example.org` on the same TCP connection to that address.

If any of the sites grouped together for the purposes of `pconn-ip` declare a `max-conns` hint, the `max-conns` value for that address is considered to be the maximum of the declared `max-conn` hints present.

5.3. ip-balance

- o Browser Hint Name: `ip-balance`
- o Description: When present, this hint indicates a preferred policy for clients to handle a DNS lookup that return multiple IPv4 addresses for the site.
- o Value Type: string
- o Contact: mnot@mnot.net

Defined values include:

- o `round-robin` - Use each IP address in succession, using the next address each time a new connection is opened.
- o `random` - Use a random IP address from the list for each new connection.
- o `failover` - Use the first IP address, falling back to the following address upon failure, and so forth.
- o `fastest` - Attempt to connect to all IP addresses, using the fastest for this and subsequent connections.

5.4. connect-timeout

- o Browser Hint Name: `connect-timeout`
- o Description: When present, this hint indicates how long the site wishes browsers to wait for a connection to be established, in seconds, before considering that connection unresponsive.
- o Value Type: integer
- o Contact: mnot@mnot.net

5.5. read-timeout

- o Browser Hint Name: `read-timeout`

Nottingham

Expires December 22, 2012

[Page 6]

- o Description: When present, this hint indicates how long the site wishes browsers to wait before considering a connection unresponsive, when data is outstanding (either a response or part thereof), in seconds.
- o Value Type: integer
- o Contact: mnot@mnot.net

Note that requests on timed-out connections can be retried, subjects to the constraints of HTTP.

5.6. max-pipeline-depth

- o Browser Hint Name: max-pipeline-depth
- o Description: When present, this hint indicates the maximum number of pipelined requests per connection that the site would like clients to use.
- o Value Type: number
- o Contact: mnot@mnot.net

5.7. small-hdrs

- o Browser Hint Name: small-hdrs
- o Description: When true, this hint indicates that clients can omit the Accept and Accept-Charset request headers when communicating with the resource, and that they can use a shortened version of the User-Agent header.
- o Value Type: prefixlist
- o Contact: mnot@mnot.net

5.8. relative-referer

- o Browser Hint Name: relative-referer
- o Description: When true, this hint indicates that servers prefer a relative URI in the Referer request header.
- o Value Type: true | false
- o Contact: mnot@mnot.net

5.9. chunk-req-bodies

- o Browser Hint Name: chunk-req-bodies
- o Description: When true, this hint indicates that the server can successfully process a request with a chunk-encoded body; i.e., that it should not return a 411 Length Required status. Note that clients may still require a 411 response status, even in when this hint is present. When false, the server may or may not require a Content-Length on requests with bodies.

- o Value Type: true | false
- o Contact: mnot@mnot.net

5.10. omit-cookies

- o Browser Hint Name: omit-cookies
- o Description: When true, this hint indicates that cookies [[RFC6265](#)] can be omitted in requests.
- o Value Type: prefixlist
- o Contact: mnot@mnot.net

6. Security Considerations

TBD

7. IANA Considerations

7.1. The 'browser-hints' Well-Known URI

This document defines the "browser-hints" Well-Known URI [[RFC5785](#)].

- o URI suffix: browser-hints
- o Change controller: mnot@mnot.net
- o Specification document(s): [this document]
- o Related information:

7.2. The HTTP Browser Hints Registry

This document establishes the HTTP Browser Hints Registry.

New hints are registered First Come First Served (see [[RFC5226](#)]), by sending e-mail to <<mailto:iana@iana.org>> (or using other mechanisms, as established by IANA).

Registration requests MUST use the following template:

- o Browser Hint Name: [name of hint]
- o Description: [description of hint]
- o Value Type: [JSON value type]
- o Contact: [e-mail address(es)]
- o Specification: [optional; reference or URI to more info]

New hints MUST be optional; they cannot place requirements upon implementations.

Likewise, new hints MUST be relevant to browser use cases; other non-

browsing hints and metadata would make the hints response undesirably large. However, note that non-browser clients MAY use them.

Finally, new hints MUST NOT make communication non-conformant with HTTP itself; i.e., this is not a mechanism for changing the HTTP protocol in incompatible ways. For example, if a hint indicates that browsers can compress request headers using GZIP, intermediaries that are interposed are likely to fail.

The initial contents of the registry are defined in [Section 5](#).

[8](#). References

[8.1](#). Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", [RFC 2616](#), June 1999.
- [RFC4627] Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", [RFC 4627](#), July 2006.

[8.2](#). Informative References

- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 5226](#), May 2008.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", [RFC 5246](#), August 2008.
- [RFC5785] Nottingham, M. and E. Hammer-Lahav, "Defining Well-Known Uniform Resource Identifiers (URIs)", [RFC 5785](#), April 2010.
- [RFC6265] Barth, A., "HTTP State Management Mechanism", [RFC 6265](#), April 2011.

[Appendix A](#). Acknowledgements

Thanks to Mike Belshé, Artur Bergman, Poul-Henning Kamp, Anirban Kundu, Patrick McManus, Steve Souders, and Martin Thompson for their suggestions and feedback.

The author takes all responsibility for errors and omissions.

Author's Address

Mark Nottingham

Email: mnot@mnot.net

URI: <http://www.mnot.net/>